Covid Analysis Matteo Bracco

Introduction

In this notebook we look for data which can be analyzed through the Lattice Gas Cellular Automata (LGCA) model, for the model details we refer to Schneckenreither et al..

The LGCA model addresses the spatial diffusion component in an epidemic spread, distributing all individuals from the population on a lattice. Infections are limited to individuals which share the same cell in the same time steps. A set of partially stochastic rules governs the motion between cells of the lattice.

To simulate an LGCA model, we will need data coming from a small and possible sparse population, to be able to work from my Laptop, and to make the spatial diffusion effect noticeable on the epidemic spread. A good possibility is to use data coming from low density areas, for instance some county in Alaska will do. We will firstly try to use those data to estimate parameters for the SIR model, then we will run a SIR and LGCA simulation.

We will then use β with the other SIR parameters to run an LGCA and some Modified LGCA simulations in a Jupyther Notebook, and we will compare the results with SIR simulations and real data here on this R Notebook.

Let's start by implementing the necessary library, importing the data and define some functions to visualize time series with ggplot2. The data selected comes from the Kodiak Island Area in Alaska. Its population from 2020 found in censu.gov was of 13100 people.

```
library("zoo")                          # For rollmean
library("ggplot2")                      # For nice time series plots
library(deSolve)                        # To solve SIR ODEs
library("reticulate")                   # To import pyhton files
library(tidyr)                          # To plot time series with
library(dplyr)                          # ggplot2
library("gridExtra")                    # to arrange ggplot2 objects into grids


# This function allows to plot obs against date, where
# obs is the collection of new registered infections, with respect to the dates
# stored in data

# Time series with only one observation
infected_series=function(date,obs,Title="Time Series of Daily New Positives"){
  df=data.frame(date,obs)
  p=ggplot(df, aes(x = as.Date(date), y = obs)) +
  geom_line()+
  xlab("date")+
    ylab("Infected")+
    labs(title=Title)
  return(p)
}

# Time series with multiple observations
```

```
Multiple_series=function(df,Title="New Infected per day"){
  di <- df %>%
  pivot_longer(cols = -times, names_to = "Infected", values_to = "Infected_Obs")

  p=ggplot(di, aes(x = as.Date(times), y = Infected_Obs, color = Infected)) +
  geom_line(size = 1)+
  labs(title = Title,
       x = "Date", y = "Value", color = "Observation") +
  theme_minimal()
  return(p)
}
```

Here is the database data for Covid-19 and the Kodiak Island selection.

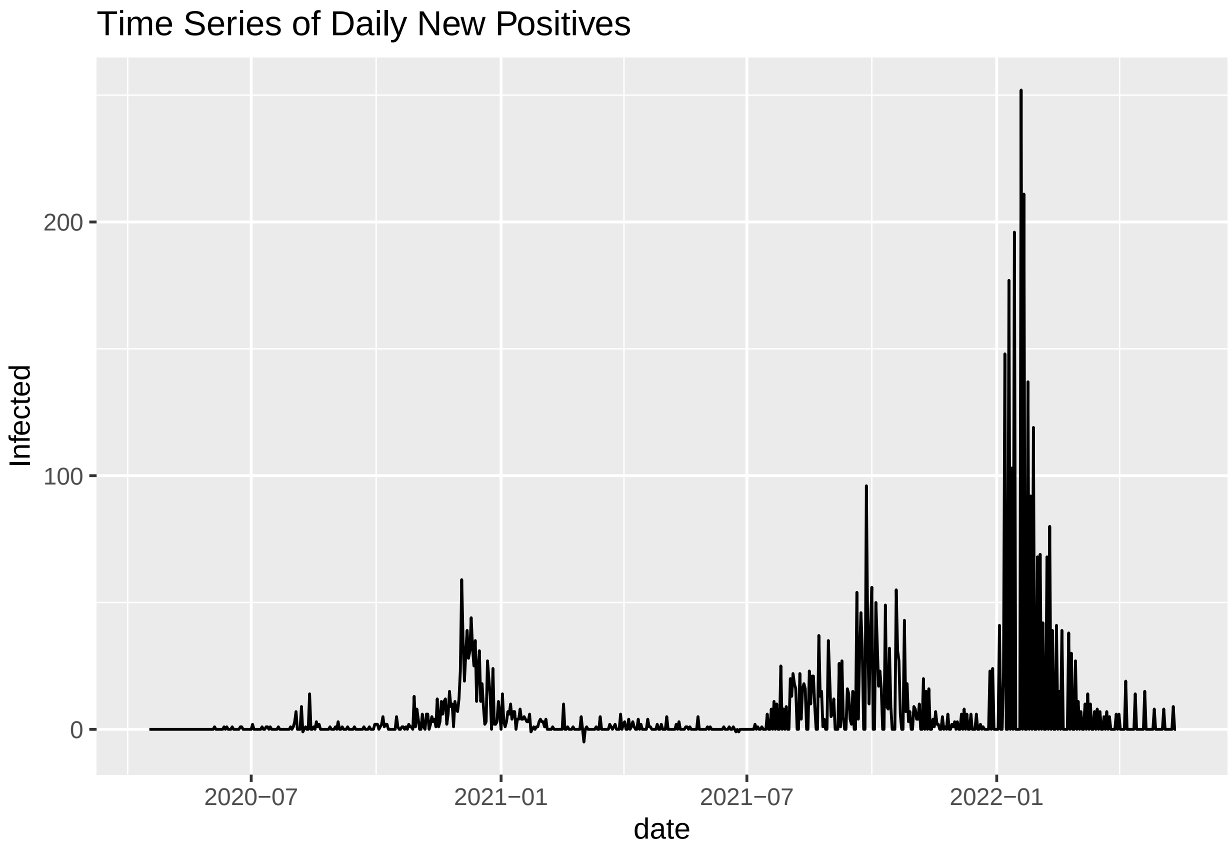
```
infections=read.csv("epidemiology.csv") # Covid Data
Inf_AL=subset(infections, location_key == "US_AK_02150")
```

Estimating Parameters for our simulations

Estimating β

We start by plotting the registered infected population over time.

```
infected_series(Inf_AL$date,Inf_AL$new_confirmed)
```



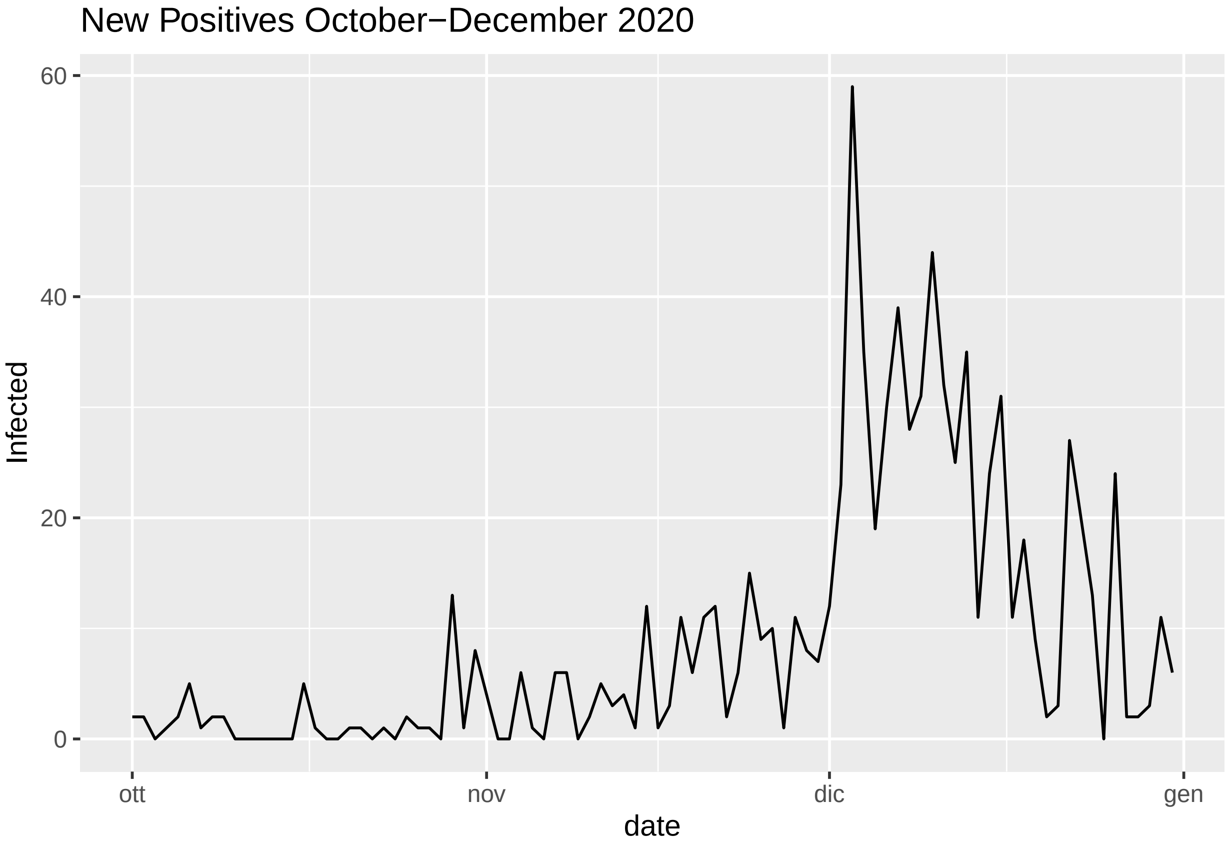
We can see that there are three separate epidemic peeks. Note that for the last two peeks, there is a big heterogeneity in the registration of new infected, represented by the big jumps in our time series, this can be caused by different flows in the data collection procedure, for instance all the new cases were registered at the end of the week instead of being registered day by day.

This pehonomenon is also present in the first peak, but it's less impactful, which probably means that the lower number of registered people can be linked to the lower number of tested people, unfortunately this is not an available data for this dataset.

We will then try to work on the first peak, as zero values would be an issue when estimating β.

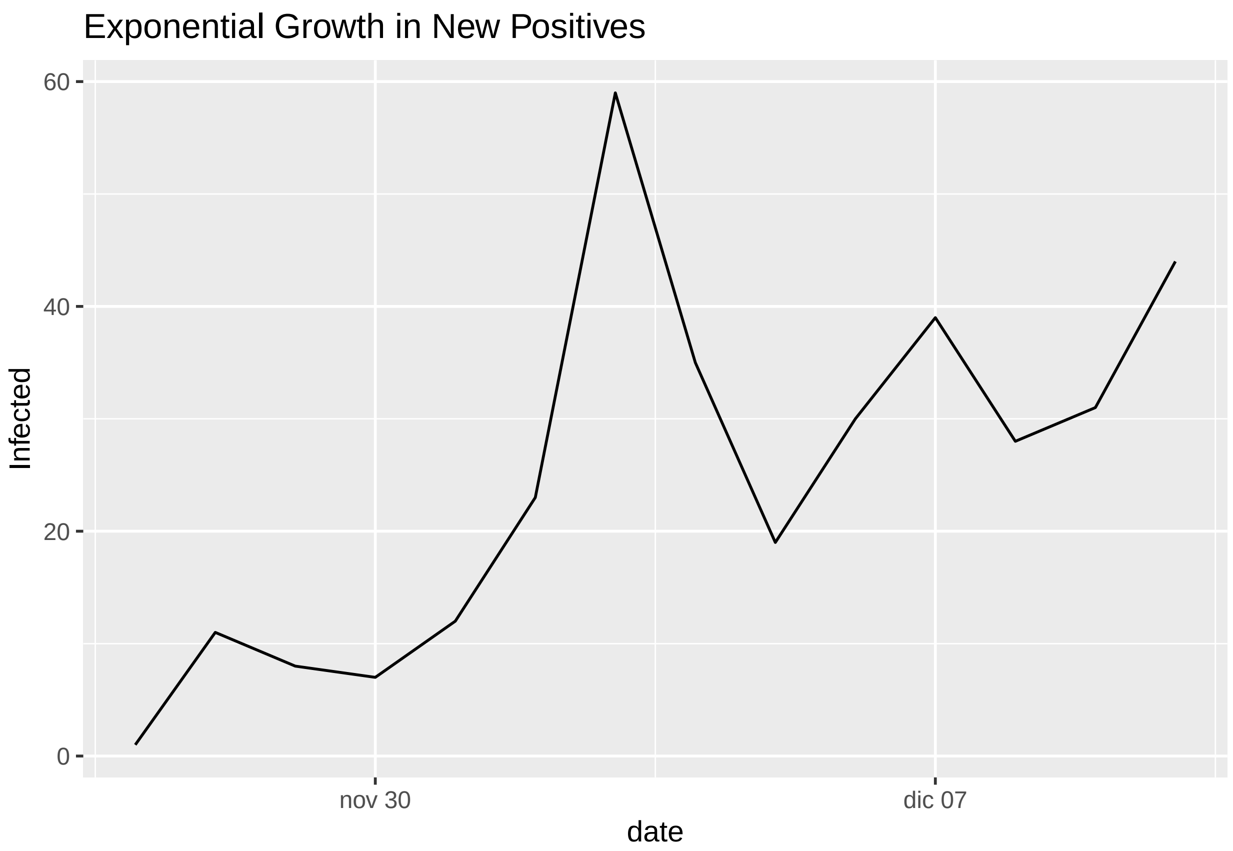
For starting, let's visualize the infected registrations from the 10^{th} of November 2020 to the 31^{st} of December 2020.

```
start=which(Inf_AL$date=="2020-10-01")
stop=which(Inf_AL$date=="2020-12-31")
obs=Inf_AL$new_confirmed[start:stop]
date=Inf_AL$date[start:stop]
infected_series(date,obs,"New Positives October-December 2020")
```



We choose a reasonable time interval as the exponential growth period of the new confirmed cases, it is difficult to accurate locate the exact time period in which the growth is exponential with the structure of our dataset

```
start2=which(date=="2020-11-27")
stop2=which(date=="2020-12-10")
date_exp=date[start2:stop2]
obs_exp=obs[start2:stop2]

infected_series(date_exp,obs_exp, "Exponential Growth in New Positives" )
```



We can then fit this data to estimate β, see Junling Ma. To understand how we will estimate β let's refresh the SIR equations. Let N be the population size, then

$$\frac{dS}{dt} = -\frac{\beta}{N}SI$$

$$\frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

In the beginning of the epidemic curve, we can suppose $S \simeq N$. Let $C = \frac{\beta}{N}SI$ be the incidence number, i.e the new cases per day, then we have $C \simeq \beta I = \beta I_0 e^{\lambda \cdot t}$, where $\lambda = \beta - \gamma$. We can fit the new confirmed cases in the exponential growth period to estimate β. Note that, even if we only register a fraction $p \in (0,1)$ of the real new positive cases, the fitting process will not change the β estimation:

$$pC \simeq p\beta I_0 e^{\lambda \cdot t} = C_0 e^{\lambda \cdot t}$$

If pC is growing exponential, we can fit the log of our data which will be growing linearly, precisely

$$log(pC) \simeq K_0 + \lambda t$$

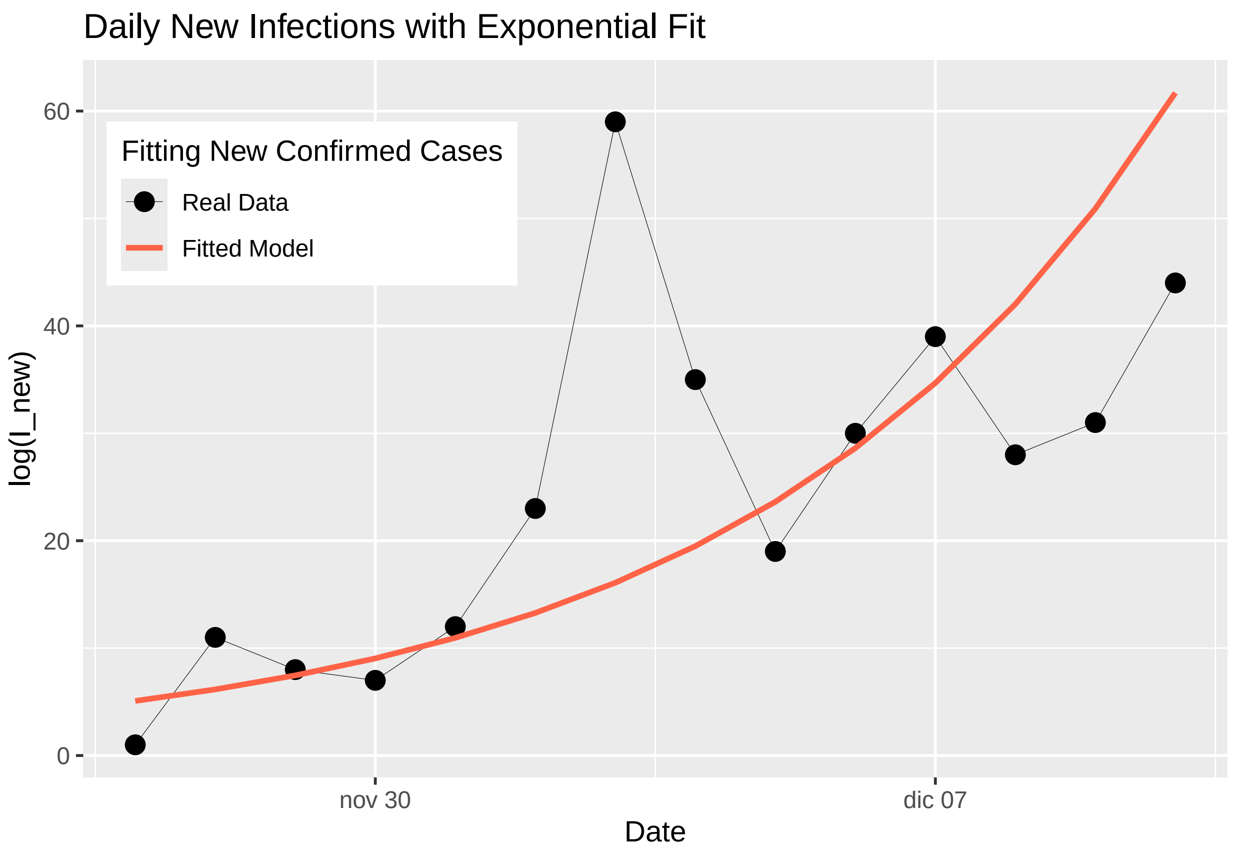
where $K_0 = log(C_0)$.

```
n=length(date_exp)
t=c(1:n)
log_obs=log(obs_exp)
Fit= lm(log_obs ~ t)
lambda <- coef(Fit)[[2]]
```

```r
K0=coef(Fit)[[1]]
gamma=1/6.5
beta=lambda+gamma
summary(Fit)
```

```
## 
## Call:
## lm(formula = log_obs ~ t)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.62578 -0.31753  0.05835  0.44151  1.29955 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)   
## (Intercept)   1.4337     0.3985   3.598  0.00366 **
## t             0.1920     0.0468   4.103  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.706 on 12 degrees of freedom
## Multiple R-squared:  0.5838, Adjusted R-squared:  0.5491 
## F-statistic: 16.83 on 1 and 12 DF,  p-value: 0.001465
```

```r
df_fit <- data.frame(t = t, new_confirmed = obs_exp)

C0=exp(K0)
# Add fitted values from the model
df_fit$fit <- exp(predict(Fit))
# Plot log-transformed data and linear fit
ggplot(df_fit, aes(x = as.Date(date_exp))) +
  geom_line(aes(y = obs_exp, color = "Real Data"), size = 0.1) +  # Real Data line
  geom_point(aes(y = obs_exp, color = "Real Data"), size = 3)+
  geom_line(aes(y = fit, color = "Fitted Model"), size = 1) +      # Fitted Model line
  labs(title = "Daily New Infections with Exponential Fit",
       x = "Date",
       y = "log(I_new)") +
  scale_color_manual(name='Fitting New Confirmed Cases',
                     breaks=c('Real Data', 'Fitted Model'),
                     values=c('Real Data'='black', 'Fitted Model'='tomato'))+
  theme(legend.position=c(0.2,0.8))
```

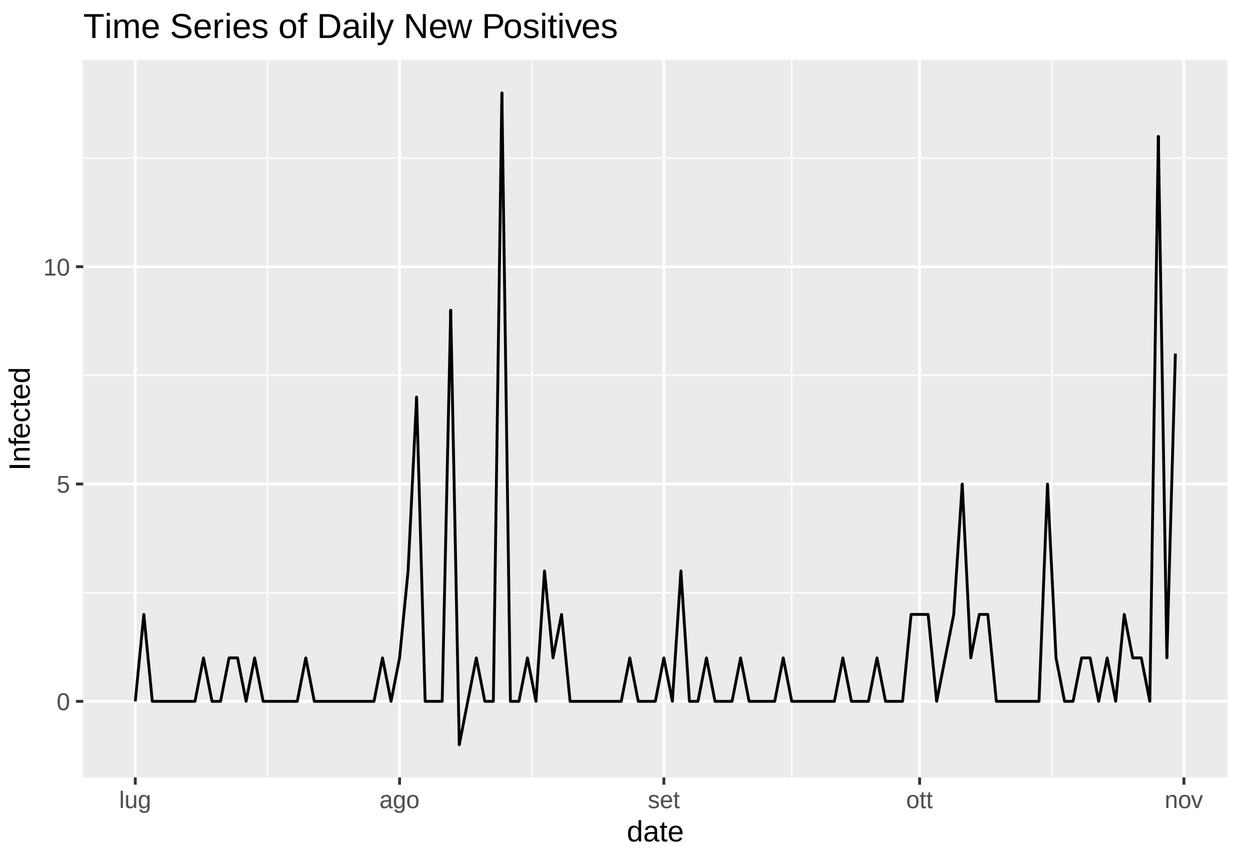


This is the value of β we will use in the simulations

The Initial Number of Infected People and the Starting Date

We now need to understand when the epidemic started and the initial number of infected people. To do so we can visualize the number of new infected in the months priors to November

```
start=which(Inf_AL$date=="2020-07-01")
stop=which(Inf_AL$date=="2020-10-31")
obs=Inf_AL$new_confirmed[start:stop]
date=Inf_AL$date[start:stop]
infected_series(date,obs)
```



As we can see there was a small peak around August, afterwards the situation seemed to be stable with almost no new positive cases between September and the start of October. Hence we can reasonably start our simulation at the beginning of October. Note that previously there weren't any large wawes of Covid-19 in the area, so we can assume the initial value of recovered people to be 0 as well.

The other option would be to simulate the epidemic from the start of July, however the β estimation has a local character. Hence it is best to limit our simulations to a span of 100 days across the autumn/winter season in Alaska, which also keeps reasonable the computational time for the simulations.

The number of infected people at October 1^{st} is the most difficult parameter to estimate, we can start by evaluating the sum in the 5 preceding days of the registered infected individuals

```
end=which(Inf_AL$date=="2020-10-01")
start=end-5
sum(Inf_AL$new_confirmed[start:end])
```

```
## [1] 5
```

This estimate is probably a lower bound on the real number of infected people for October $1^{st}. We can at least triple such number to represent the initial population of infected people. However if the models are robust enough, we shall not expect small changes in the initial values to compromise at least the quality development of our simulations.

Parameters for the Simulations

We can the visualize the initial values and parameters for our LGCA and SIR simulations below.

```
startN=which(Inf_AL$date=="2020-10-01")
endN=which(Inf_AL$date=="2021-01-09")
```

```
I0=15
date0="2020-10-01"
N=13100
gamma=1/6.5

Parameter=c("Starting Date","Total Population","Initial Population of Infected People","gamma","beta")
Value=c(date0,N,I0,gamma,beta)
Initial=data.frame(Parameter,Value)
Initial
```

```
##                                  Parameter             Value
## 1                            Starting Date        2020-10-01
## 2                         Total Population             13100
## 3 Initial Population of Infected People                15
## 4                                    gamma 0.153846153846154
## 5                                     beta  0.34588132363107
```

Simulations Comparisons

We load the LGCA simulation results from the Jupyter Notebook, we will compare this simulation with real data and an SIR simulation.

```
history=py_load_object("LGCAv4.pckl")
```

SIR Model

We then need to evaluate a classic version of the SIR model, we will use the simplest version as it is the one on which the LGCA model is built on.

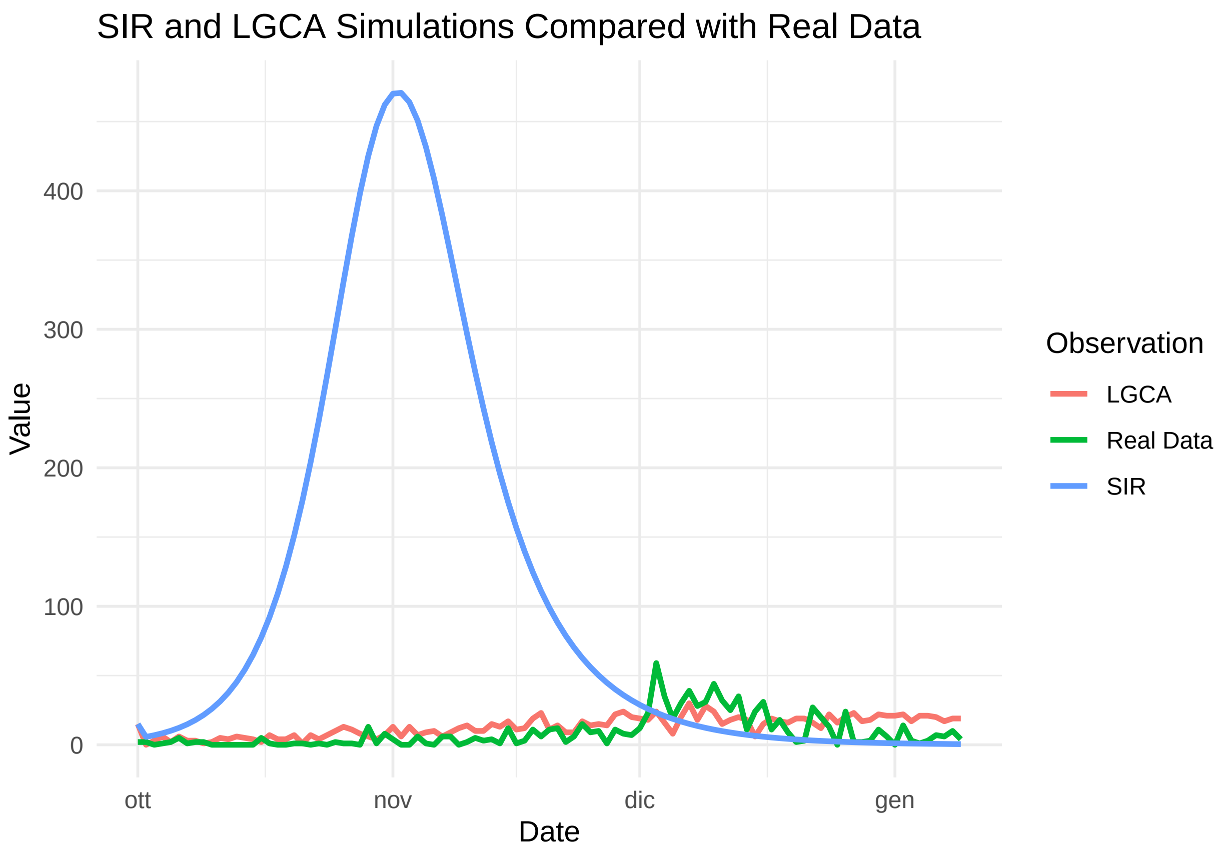
```
sir_equations <- function(time, variables, parameters) { # SIR equation
  with(as.list(c(variables, parameters)), {
    dS <- -B * I * S
    dI <-  B * I * S - G * I
    dR <-  G * I
    return(list(c(dS, dI, dR)))
  })
}

parameters_values <- c( # Params for SIR
  B  = beta/N,
  G = gamma
)

initial_values <- c( # Intial Values
  S = N-I0,
  I =   I0,
  R =   0
)

time_values <- seq(0, 100) # 100 days

SIR_obs =as.data.frame(ode( #Solvin the ODE
  y = initial_values,
  times = time_values,
  func = sir_equations,
  parms = parameters_values
))
```

We can derive the incidence of the susceptibles with the following functions, an analogous version can be found in the Python script

```
find_incid <- function(I,R){
  n <- length(I)
  incid <- c(15)
  for (i in 2:n) {
    I_today <- I[i]
    R_new <- R[i] - R[i-1]
    I_yest <- I[i-1] - R_new
    I_new <- I_today - I_yest
    incid <- c(incid, I_new)
  }

  return(incid)
}

incid_SIR=find_incid(SIR_obs$I,SIR_obs$R)
```

We can then build a data.frame containing real data, LGCA data and SIR data to compare the results

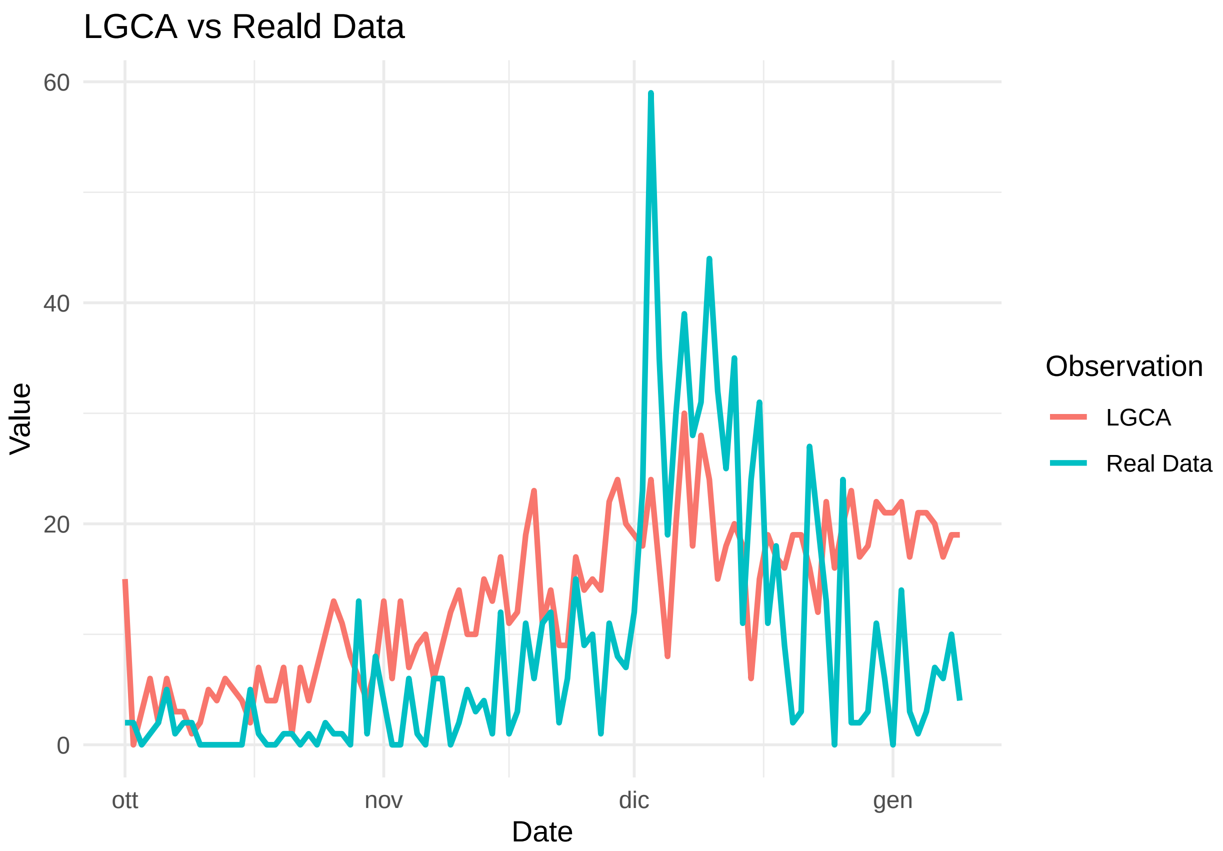
```
start=which(Inf_AL$date==date0)
stop=start+100
times=Inf_AL$date[start:stop]
incid_real=Inf_AL$new_confirmed[start:stop]
Data_Incid=data.frame(times,incid_SIR,history,incid_real)
colnames(Data_Incid) <- c("times","SIR","LGCA","Real Data")
Multiple_series(Data_Incid,"SIR and LGCA Simulations Compared with Real Data")
```



The main issue we are trying to address with the LGCA model is quite noticeable from this simulation: the real peak of new infected people is shifted in time with respect to the SIR estimates.

Now, we would like to see the peak of the LGCA estimate near to the peak of the real data, however we do not even see a peak for the LGCA simulation from this visualization. Let's remove the SIR data to have a better view of the LGCA results

```
Incid_new=data.frame(times,history,incid_real)
colnames(Incid_new) <- c("times","LGCA","Real Data")
Multiple_series(Incid_new, "LGCA vs Reald Data")
```



The LGCA model seems in fact to have a very low and very large peak 20 days after the SIR model, quite closer to the real registered peak, but the estimates of the new positive cases are very low, even when compared with the registered data, and the peak is barely noticeable, hence we can't really exclude that it only happened because of stochastic effects of this specific simulation. Obviously the best solution would be to run more simulations and take the average to solve such issue, but we do not have the computational power to do so.

Nevertheless, the development of the LGCA simulation doesn't seem to follow the real epidemic spread, even from a qualitative point of view, as we should at least see a significant increase in the new registered cases.

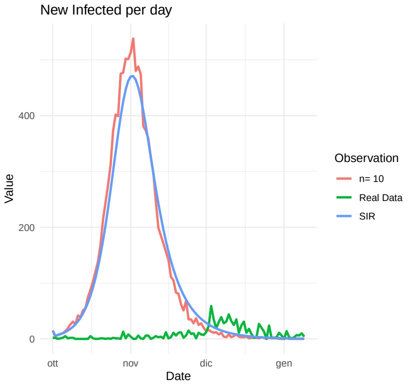
This is probably due to the restrictive law of motions and the large grid imposed by the LGCA model.

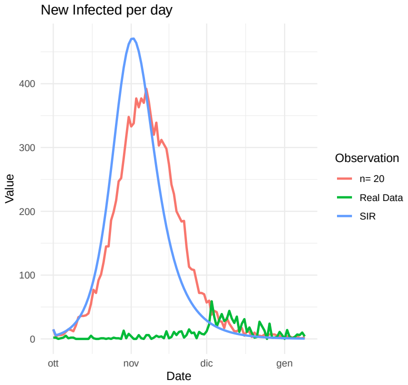
However, such restrictions where not justified theoretically in Schneckenreither et al., hence we may as well change them. The Modified LGCA algorithms removes the restriction imposed by the law of motions between neighbor cells, as well as removing the "PacMan Effect". More details on the LGCAìs models are available in the Jupyter Notebook.

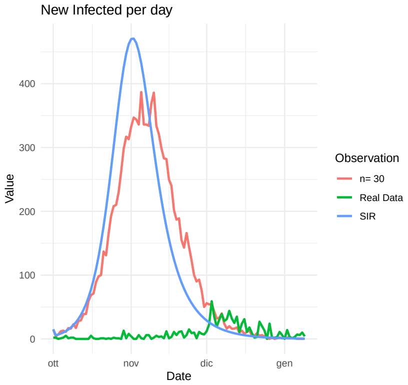
Moreover we allow for smaller squared grids which also drastically decrease the computational time for each simulation. We can for example compare the effects of this Modified LGCA algorithm on squared grid of dimension $10k, \quad k = 1, \dots, 10$.

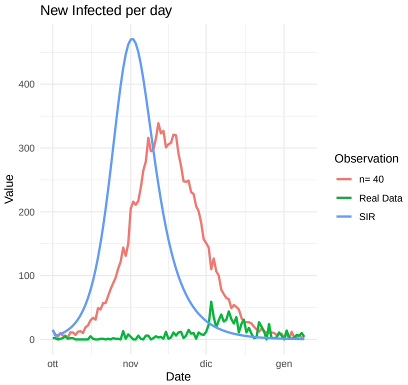
```
new_history=py_load_object("modified10v4.pckl")
N=length(incid_SIR)

pad <- function(x, len) {
  c(x, rep(0, len - length(x)))
}
```

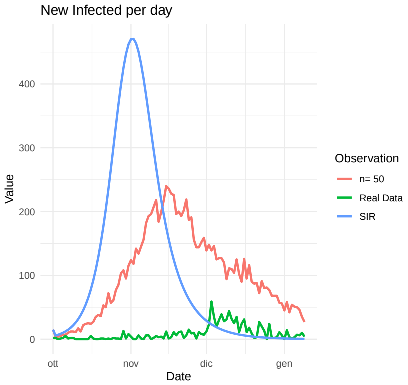
```r
n=10
plots=c()
i=0
for(simul in new_history){
  i=i+1
  third_lab=paste("n=",n)
  pad_sim=pad(simul,N)
  Modified_data=data.frame(times,incid_real,pad_sim,incid_SIR)
  colnames(Modified_data) <- c("times","Real Data",  third_lab,"SIR")
  p=Multiple_series(Modified_data)
  n=n+10
  plots[[i]]=p
}



grid.arrange(grobs = plots, ncol = 3)
```

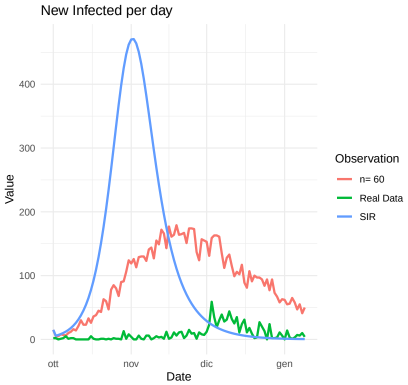


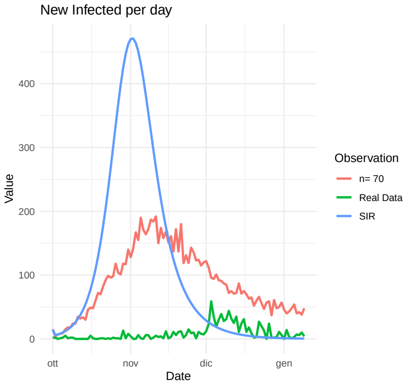


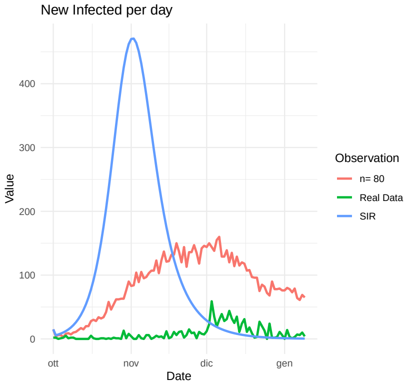


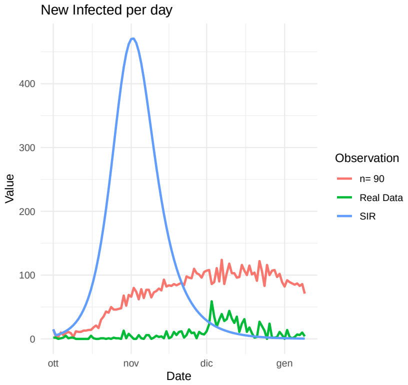


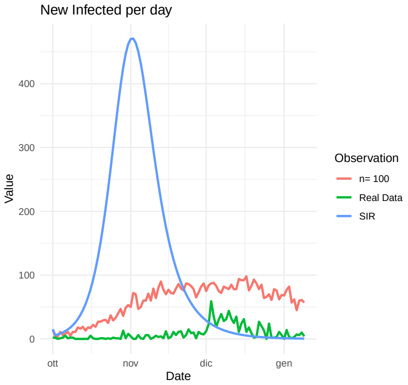


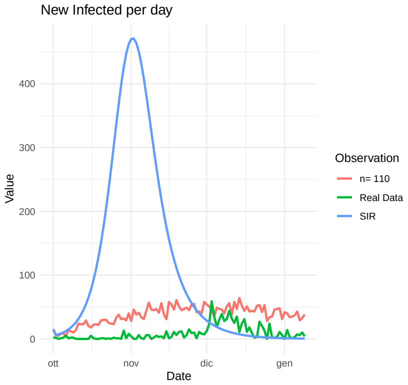


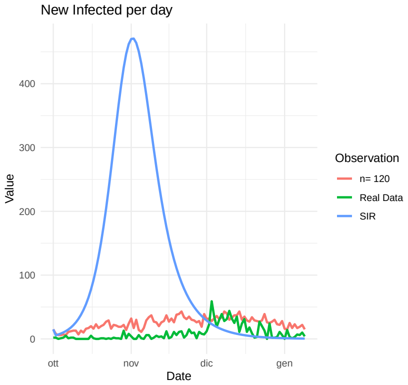




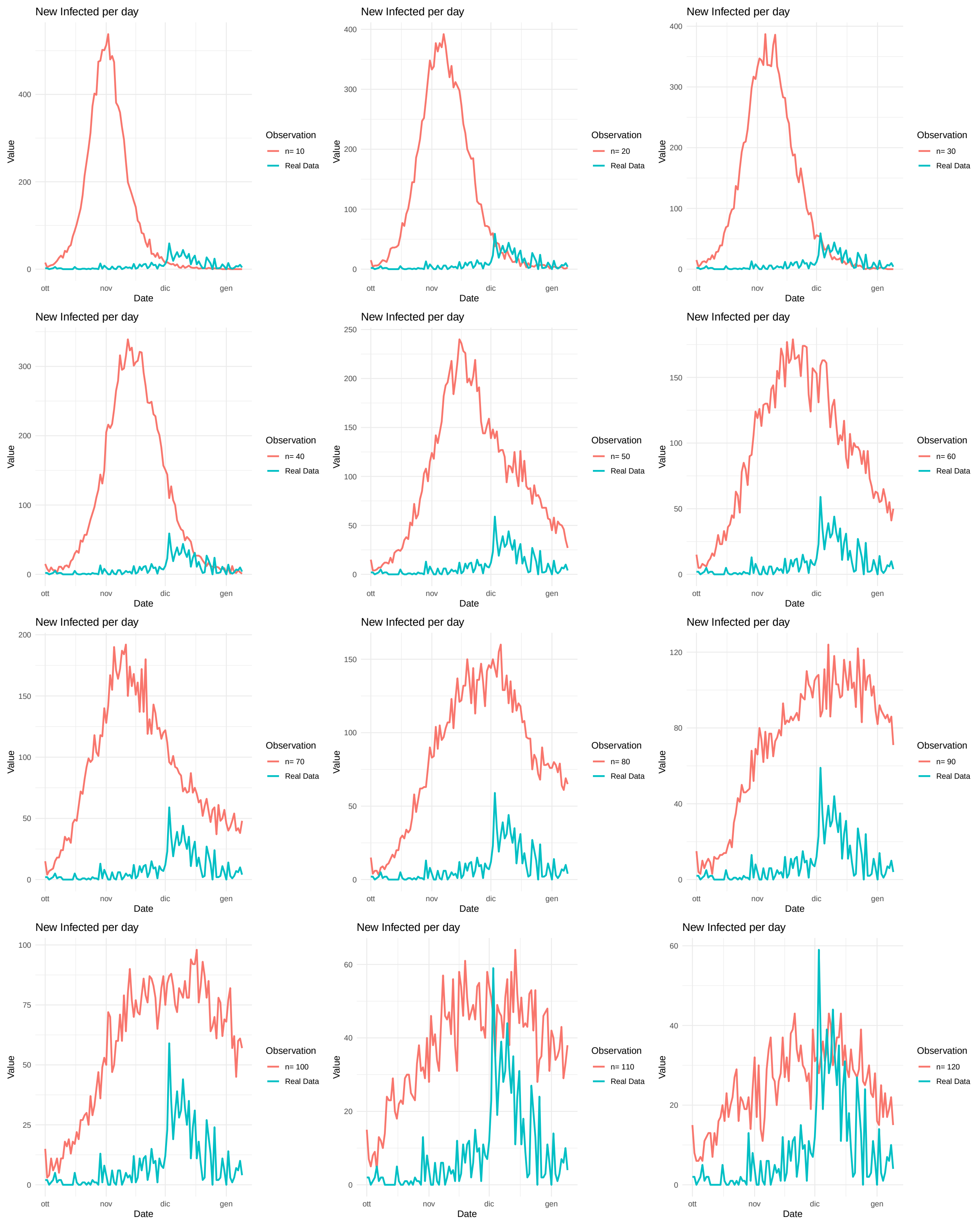








```r
n=10
plots=c()
i=0
for(simul in new_history){
  i=i+1
  third_lab=paste("n=",n)
  pad_sim=pad(simul,N)
  Modified_data=data.frame(times,pad_sim,incid_real)
  colnames(Modified_data) <- c("times",  third_lab,"Real Data")
  p=Multiple_series(Modified_data)
  n=n+10
  plots[[i]]=p
}

grid.arrange(grobs = plots, ncol = 3)
```



The modified LGCA simulations presents some interesting features linked to the base dimension d of the lattice,

When $d = 10$ the simulation seems to behave accordingly to the SIR simulation. When we start increasing the lattice dimension, we see a shift in the peak of the simulation, both down and right. This is probably due to the spatial diffusion becoming a factor: individuals in cells further a part from initial outbreaks are not subjected to the virus in the initial stage of the epidemic.

The peak seems to match the real data peak around $d = 80, 90$, but we should take this piece of information with caution, as we can't really be sure on the starting day of the epidemic from real data.

However, there is certainly a delay between the peaks of SIR and real data, which seems to be at least partially correctable through the Modified Lattice Model.

Note that different laws of motions, and initial distributions of the population on the lattice may results in different results in the qualitative behaviour of the Modified LGCA simulations.

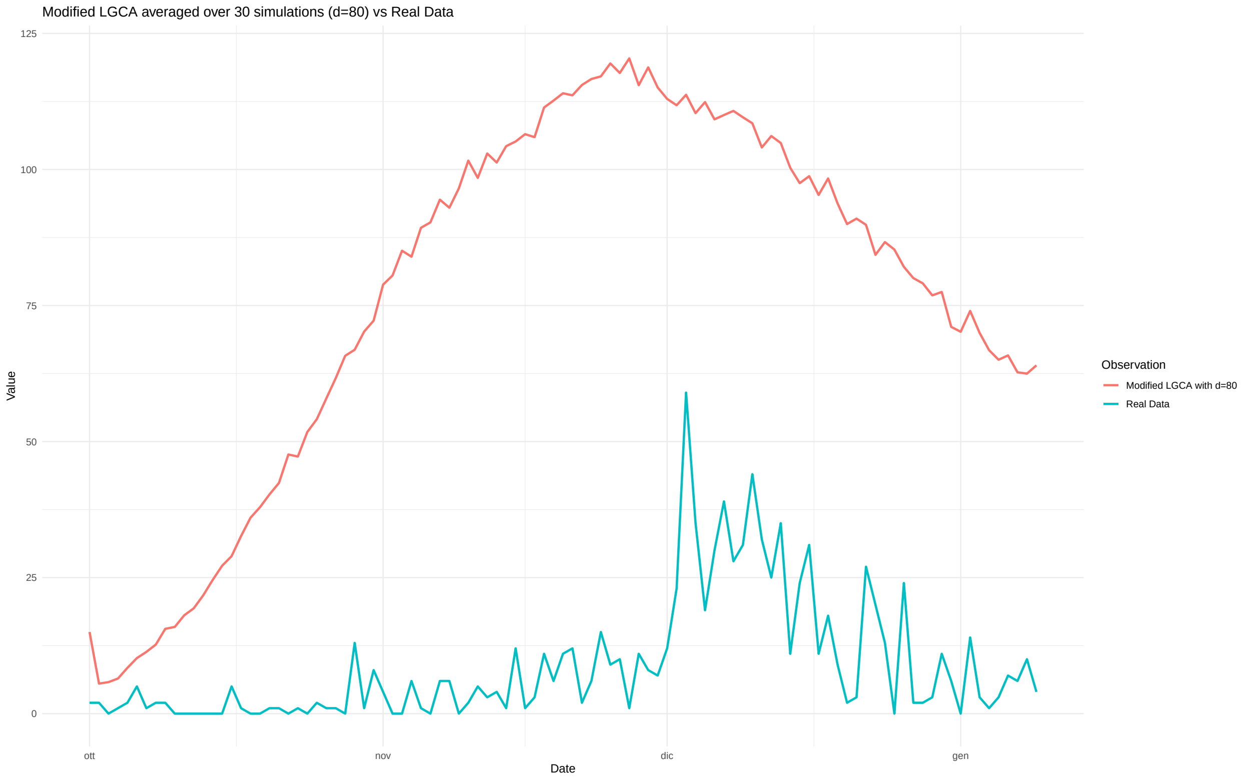
Once d becomes too large, the peak seems to flatten out and the trend of new positives seems to be rather stable in time, similarly to what we saw in the original LGCA model. We do not know if the peak will arrive later on in time, but it seems quite unlikely. This suggests that the biggest difference between the two models resides in the grid dimensions rather then the different laws of motions.

In conclusion, the results of the Modified LGCA simulation seems to suggest that incorporating correctly the spatial diffusion component through a lattice representation may help to explain the qualitative behaviour of an epidemic. Moreover, these results leave open different questions on the dynamical system described by the LGCA modification algorithm, which to me seems important to better understand how spatial diffusion can be modeled in epidemic spreads:

- Can we define an optimal value for the dimension d of the grid in the Modified LGCA simulation? Should it depend on the individuals habits, the population density and the geographical structure of the area we are studying, and the other parameters (β, γ) of our model?
- Is the dynamical systems sensitive to the initial value of positive people I_0? At least for small changes in small values of I_0, is the system stable? Which are the equilibrium points of the system? Are there periodic solutions which may arise in real scenarios?
- How does the peak of infected people relates to the lattice dimension? Does the peak flattens out for $d \simeq \sqrt{N}$ or does it appear later on in time? Is it possible to find a formula to link the peak of infected people, for instance $t_{max} = argmax_{t \in T} C(t)$ to the model parameters, especially to the grid dimension d?
- Can we introduce different law of motions to incorporate lockdowns or restriction to movement? What happens if we change the initial distribution of Infected people on the lattice? Could we use different representation for the lattice (e.g rectangular), to better model the geography of the area we are analyzing?

To end the notebook, let's run 30 simulations with dimension $d = 80$, to balance out the stochastic effects. We can compare the results with real data, keep always in mind that we are not sure about the starting date and the initial conditions of the epidemic

```r
v40=py_load_object("v40.pckl")
v40_df=as.data.frame(v40)
means40=rowMeans(v40_df)
Mod40=data.frame(times,incid_real,means40)
colnames(Mod40)=c("times","Real Data","Modified LGCA with d=80")
Multiple_series(Mod40,"Modified LGCA averaged over 30 simulations (d=80) vs Real Data")
```



References

- Schneckenreither, Gunter and Popper, Nikolas and Zauner, Gunther and Breitenecker, Felix. *Modelling SIR-type epidemics by ODEs, PDEs, difference equations and cellular automata–A comparative study.* Simulation Modelling Practice and Theory 16(8)1014-1023 2008. https://doi.org/10.1016/j.simpat.2008.05.015

- Junling Ma. *Estimating epidemic exponential growth rate and basic reproduction number* Infectious Disease Modelling 5:129-141 2020. https://doi.org/10.1016/j.idm.2019.12.009